

1000-1010 UP
1030-1040 DOWN

e.g. In the 'WEST' routine, create line:
925 IF LO = 1 THEN LO = 22

and in the 'EAST' routine, create line:
975 IF LO = 22 THEN LO = 1

Now, with some thought, a full 16K custom-made adventure can be written from this 'skeleton' adventure.

The SAVE & LOAD routines in the program were written for disk based micros. Owners of tape bases systems will need to make the following modifications:

DELETE LINES 1620-1650 AND
1700-1730.

NOW INSERT THESE LINES ...

**** SAVE ROUTINE ****

```
1620 C$ = " ": FOR I9 = 1 TO OB:
      C$ = C$ + STR$( B(19) ) + " ":
      NEXT I9
1630 PRINT # - 1, C$, LO
1640 RETURN
```

**** LOAD ROUTINE ****

```
1700 INPUT # - 1, C$, LO: IN = O:
      D$ = " "
1710 FOR I9 = 1 TO OB
1720 IN = IN + 1: M$ = MID$( C$, IN,
      1): D$ = D$ + M$
1730 IF M$ = "/" THEN D$ = LEFT$(
      D$, LEN(D$) - 1):
      B(I9) = VAL(D$): D$ = " ": GOTO
      1740 ELSE 1720
1740 NEXT I9
1750 RETURN
```

BATTLESHIPS (VZED 8K)

This is the old board game of Battleships and cruisers. The screen is divided into a 9 x 9 grid. The computer 'hides' a total of 10 ships at random around this grid. There are four types of ships — 1 Battleship which occupies four adjacent squares, two Cruisers which occupy three adjacent squares each, three destroyers which occupy two adjacent squares each and four submarines occupying yes, you've got it, one square each.

You must enter the coordinates of a square in the grid, at which time the computer prints either a letter in that square, denoting the type of vessel hit, or will print an asterisk if the square is empty. The object of the game is to sink all the vessels with the least possible number of shots. Good hunting!

JUNIOR MATHS (VZED 8K)

This program tests the four basic mathematical functions: Addition, Division, Subtraction and Multiplication. Whilst not an educational program in the strictest sense, it does serve to reinforce lessons already learnt. You are first asked to choose the type of problem after which a graphics screen is presented with an area for the questions and answers and a representation of a persons head with a non-committal expression and some ominous blue water at the bottom. 10 questions are

presented one at a time. A correct answer is rewarded by a smile and some uplifting music whilst an incorrect answer causes a frown and depressing music. In this event, the correct answer is also displayed. When the ten questions have been presented, your score and percentage correct are shown.

Now comes the odd bit which may cause our mailbags to bulge with irate letters from outraged child psychologists. In the original version, the author "punished" an imperfect score by raising the water level until it covered the head. He soon found that children using it would deliberately enter incorrect answers just to see this happen. So he reversed the procedure. Now to submerge the hapless head, one must get a perfect score! By the way, the level of difficulty is appropriate to children aged from 9-11.

DISK DIRECTORY PROGRAM (48K/MOD III DISK)

by Ross Smith

REQUIREMENT TO RUN PROGRAM

A 32K or 48K TRS-80 Model III with at least one disk drive. A second drive simplifies the entering of data. A printer is optional. The program was written to be used with TRSDOS 1.3 and will only operate under other operating systems if lines 10 to 30 are modified. These lines use a call to a TRSDOS I/O call (\$RAMDIR — 4290H) which is documented in the TRSDOS owner's manual.

DESCRIPTION OF PROGRAM

This program was written to enable the user to keep track of his disk programs. It will maintain a catalog of the name of the program, the extension and the name of the diskette on which the program is stored. The program has been automated as far as possible including the use of INKEY\$. The only data that the user needs to enter is the program's name, as the other relevant data is automatically read off the diskette by a machine language subroutine.

The data is stored as linked lists in such a way that all three lists of data can be sorted simultaneously. The data can then be stored in its sorted form on diskette. Thus, although the actual sort can take several minutes, it only needs to be carried out once after new data has been entered into the file.

The program protects enough memory to hold a short machine language program as well as a full diskette directory when it is read from a disk by the TRSDOS I/O call \$RAMDIR. As this is done from within the program there is no need to remember to set the memory size before using it.

Several options have been included in this program to allow maximum flexibility and ease of use. The following summarises these options:

(1) ADDING A DISKETTE — Lines 1000 to 1990

This is the fundamental part of the program and allows the contents of

up to 100 disks (up to 30 for a 32K machine) to be stored in memory. A total of 700 (300) programs can be stored at a time. After inputting the diskette's name the user is required to put the diskette in the appropriate drive and press /ENTER/. The directory is then automatically read into memory using a machine language program stored in high memory which calls a TRSDOS I/O call. The call (\$RAMDIR — 4290H) is clearly documented in the TRSDOS owner's manual. The name of each program on the diskette, its extension and the name of the diskette are stored as a linked list in array D(2,M). The linking occurs through array T(2,M) in such a way that all three lists of information can be sorted at the same time. The diskette name is also added to a separate array A(N) for later use. Before returning to the main menu this array is sorted using Disk BASIC's machine language sort CMD"O". A diskette containing Disk BASIC must be in Drive 0 when this occurs. Thus when using this program on a single drive machine ensure that a diskette containing Disk BASIC is in the drive before hitting /ENTER/ to return to the main menu.

(2) DELETING A DISKETTE — Lines 2000 to 2990

Since the data is stored as linked lists, this routine cannot simply clear the appropriate entries in the relevant array. Instead, a graphic symbol is inserted into the appropriate elements of the arrays which are then sorted. The graphic symbol is thus moved to the end of each of the three columns of the array and can be cleared. As is mentioned below this sort can take a considerable time depending on the number of elements in the array.

(3) UPDATING A DISKETTE — Lines 3000 to 3990

This part of the program uses the above two subroutines to first remove a diskette and then enter the updated version into memory. As with the previous routine this one may take considerable time due to the need to sort the data before deleting the old information.

(4) LISTING DATA — Lines 4000 to 4990

This subroutine allows the data to be listed to the video display. If the printer option is engaged (see below) the data is also sent to a printer. Four options are available. The first three list all the stored data. They differ only in which category is listed first (in alphabetical order if the list has been sorted). The fourth option lists only the diskette names. This option can be used to quickly see which names have already been used.

(5) SORTED DATA — Lines 5000 to 5990

This routine allows the data to be sorted by program name, program extension, diskette name or all three. The data is stored in array D(2,M) as three linked lists using array T(2,M) to maintain the links. The data in each of the three columns thus can be in-

BATTLESHIPS VZ 200

```

3 CLS:COLUR,1:PRINT@170,"FOR VZ-200"
4 PRINT@201,"BY R. CARSON":PRINT@235,"ADELAIDE"
5 PRINT@33,"***THE GAME OF BATTLESHIPS***":REM COPYRIGHT
6 PRINT@425,"INSTRUCTIONS?":PRINT@456,">>Y=YES  N=NO<<"
7 K$=INKEY$
8 I$=INKEY$:IF I$=""THEN8
9 IF I$="Y"THEN12
10 IF I$="N"THEN90
11 IF I$<>"Y"THEN7:IF I$<>"N"THEN7
12 CLS:PRINT"THE PLAYING AREA REPRESENTS AN ";
13 PRINT"AREA OF SEA. THE COMPUTER IS ";
14 PRINT"CONTROLLING TEN SHIPS, A BATTLE-";
15 PRINT"SHIP, 2 CRUISERS, 3 DESTROYERS ";
16 PRINT"AND 4 SUBMARINES. OF COURSE, I ";
17 PRINT"CAN'T TELL YOU WHERE THEY ARE, ";
18 PRINT"ONLY THE COMPUTER KNOWS, UNTIL ";
19 PRINT"YOU HIT THEM. THE SHIPS ARE ";
20 PRINT"DIFFERENT SIZES, AND ARE IDENTI-";
21 PRINT"FIED BY THE INITIAL LETTER. THE ";
22 PRINT"BATTLESHIP OCCUPIES FOUR SQUARES";
23 PRINT"LIKE THIS: BBBB, ACROSS OR DOWN."
24 PRINT:PRINT" PRESS <SPACE> TO CONTINUE"
25 K$=INKEY$
26 I$=INKEY$:IF I$<>" "THEN26
32 CLS:PRINT"THE CRUISERS THREE SQUARES, THE ";
33 PRINT"DESTROYERS TWO SQUARES, AND THE ";
34 PRINT"SUBMARINES ONE SQUARE, ALWAYS IN";
35 PRINT"A STRAIGHT LINE. SHIPS MAY TOUCH";
36 PRINT"OR LAY ALONGSIDE EACH OTHER.YOU ";
37 PRINT"FIRE A SHOT BY GIVING TWO ";
38 PRINT"NUMBERS. THE FIRST ON THE LEFT, ";
39 PRINT"THE SECOND AT THE TOP. IF YOU ";
40 PRINT"HIT ANYTHING, A LETTER WILL BE ";
41 PRINT"PRINTED TO TELL YOU WHICH TYPE ";
42 PRINT"OF SHIP YOU HIT. TO SINK IT, YOU";
43 PRINT"MUST HIT ALL THE SQUARES OF THAT ";
44 PRINT"PARTICULAR SHIP."
45 PRINT:PRINT" PRESS <SPACE> TO CONTINUE"
46 K$=INKEY$
47 I$=INKEY$:IF I$<>" "THEN47
52 CLS:PRINT"IF YOU MISS, THEN * IS PRINTED ";
53 PRINT"TO REMIND YOU THAT YOU HAVE SHOT";
54 PRINT"INTO THAT SQUARE BEFORE."
55 PRINT:PRINT"YOUR NUMBER OF SHOTS IS SHOWN AT";
56 PRINT"THE BOTTOM OF THE SCREEN AND THE";
57 PRINT"BEST SCORE YOU ACHIEVED DURING A";
58 PRINT"SERIES OF GAMES. THE GAME ENDS ";
59 PRINT"WHEN ALL SHIPS HAVE BEEN SUNK."
60 PRINT:PRINT:PRINT" HAPPY HUNTING"
61 PRINT:PRINT:PRINT" PRESS <SPACE> TO START"
62 K$=INKEY$
63 I$=INKEY$:IF I$<>" "THEN63
90 CLS
95 X=0
100 A=100
110 DIM G(100)
120 D=0
125 CLS:PRINT@196,"WAIT---ARRANGING FLEET"
130 C=0
140 FOR B=1 TO 100
150 G(B)=0

```

```

160 NEXT B
170 E=4
180 F=1
190 H=INT(RND(X)*2)
195 W=0
200 IF H=0 THEN J=RND(9)
202 IF H=1 THEN J=RND(4)
205 IF H=1 THEN K=RND(9)
212 IF H=0 THEN K=RND(4)
220 L=0
230 P=10*J+K
250 FOR M=0 TO (E-1)
255 IF H=0 THEN R=P+M
260 IF H=1 THEN R=P+10*M
280 IF L=0 AND G(R)<>0 THEN W=W+1
290 IF L=1 THEN G(R)=E
300 NEXT M
305 IF W>0 AND W<10 THEN 190
306 IF W=10 THEN 140
310 IF L=1 THEN 400
320 L=1
330 GOTO 250
400 F=F+1
410 IF F<4 THEN E=3
420 IF F>3 AND F<7 THEN E=2
430 IF F>6 THEN E=1
440 IF F=11 THEN 700
445 GOTO 190
450 PRINT@435,"      ":INPUTS
458 IF S<11 THEN 450
460 IF S>99 THEN 450
465 T=INT((S)/10)
470 U=S-T*10
472 IF G(S)=5 THEN 450
475 IF G(S)=4 THEN S1$="B"
480 IF G(S)=3 THEN S1$="C"
485 IF G(S)=2 THEN S1$="D"
490 IF G(S)=1 THEN S1$="S"
495 IF G(S)=0 THEN S1$="*"
500 V=U*2+T*32+101
510 PRINT@V,S1$
520 C=C+1
530 IF G(S)>0 THEN D=D+1
535 G(S)=5
540 PRINT@418,"SHOTS:";C;
550 IF D<20 THEN 450
560 IF A<82 THEN PRINT@457,"BEST SCORE:";A
570 IF C=A THEN A=C
580 PRINT@482,"ANOTHER GAME?>>Y=YES  N=NO<<";
585 K$=INKEY$
590 I$=INKEY$:IF I$=""THEN590
595 IF I$="Y"THEN120
597 IF I$="N"THENCLS:END
600 IF I$<>"Y"THEN585
610 IF I$<>"N"THEN585
700 CLS:PRINT@39,"***BATTLESHIPS***"
720 PRINT:PRINTTAB(7);"1 2 3 4 5 6 7 8 9"
730 FOR N=1 TO 9
740 PRINTTAB(4);N;". . . . . "
750 NEXT N
760 GOTO 450

```